Solution explanation :

For every block, you want to go through every requirement, and for every requirement, you want to find the closest other block with that requirement (or rather, the smallest distance to another block with that requirement). Once you’ve done that for every requirement and for every block, you want to pick, for every block, the distance of the farthest requirement. You can do this with three nested “for” loops.

Is there a way to optimize on the solution mentioned above(that uses three nested “for” loops) by precomputing the smallest distances of every requirement from every block?

For every requirement, you should be able to precompute its smallest distances from every block by doing two simple passes though the array of blocks: one pass from left to right and one pass from right to left. Once you have these precomputed values, you can iterate through all of the blocks and pick the biggest of all the precomputed distances at that block.

**Optimal Space & Time Complexity** O(br) time | O(br) space - where b is the number of blocks and r is the number of requirements.